

Truthful Complex-valued Knapsack Problem and Discrete Optimization in A/C Electrical Grid

Chi-Kin Chau

Masdar Institute of Science and Technology, UAE
Email: ckchau@masdar.ac.ae

Lan Yu

Nanyang Technological University, Singapore
Email: yula0001@ntu.edu.sg

Abstract—Since efficient power allocation is a critical requirement for smart grid, we study an important basic setting – *knapsack problem with selfish users*, whereby we design a mechanism to find a utility-maximizing allocation for a group of users with inelastic demands, such that users truthfully reveal their private utility information. As a departure from the traditional setting, complex-valued entities (e.g. power, voltage, and current) are common in A/C electrical grid. There were only few results in the literature concerning complex-valued entities for discrete optimization, because they are substantially more challenging. In this paper, we introduce a non-trivial generalization of knapsack problem with a complex-valued constraint on A/C power, which casts fundamental insight to discrete optimization for smart grid. We provide results of approximability (the existence of a $(\frac{1}{2} - \epsilon)$ -approximation algorithm) and inapproximability (the absence of FPTAS unless $P = NP$) for a class of complex-valued knapsack problem, considering complex-valued A/C power with non-negative real and imaginary parts. Further, we achieve truthfulness in this setting. We also apply our results to a setting of A/C power with moderate negative imaginary part.

Index Terms—Complex-valued Knapsack Problem, Discrete Optimization for Smart Grid, A/C Electrical Grid, Approximation Algorithms, FPTAS, Incentive Compatibility, Truthfulness

I. INTRODUCTION

The rise of smart grid provides a new breed of resource allocation problems. Unlike the traditional settings of resource allocation, many entities are expressed by complex values in A/C electrical systems, which can conveniently capture both phase and magnitude in A/C electrical systems. While real-valued entities are often studied in traditional discrete optimization, the consideration of complex-valued entities has received limited attention, despite its importance to smart grid.

In this paper, we consider a fundamental problem of utility-maximizing allocation of power in A/C electrical grid. Our study is based on the well-known *knapsack problem* [4]. In knapsack problem, there are a number of users; each has an inelastic demand of power (expressed by a complex value) and a utility value if the demand is satisfied. The goal is to find an allocation of power to a subset of users, such that the total utility of satisfiable demands is maximized. The constraint is that, each inelastic demand is either satisfied or dropped, and the total satisfiable demand is at most the capacity of supplies.

In the terminology of power systems [3], the real part of complex-valued power is known as *active* power, the imaginary part is known as *reactive* power, whereas the magnitude is known as *apparent* power. Electric appliances have different active and reactive power requirements¹, while the electrical

grid also has capacity constraints on its active, reactive and apparent power supplies. Therefore, the discrete power allocation problem in A/C electrical grid is more complicated than the traditional knapsack problem, which concerns allocating a single real-valued resource. One can formulate discrete power allocation problem with separate capacity constraints on active and reactive power as two dimensional knapsack problem [4]. However, the problem involving capacity constraint on apparent power appears as a new problem, involving a quadratic constraint, rather than linear constraints in traditional knapsack problems. In this paper, we provide the first study of complex-valued knapsack problem with two important contributions.

1. (In)approximability: Our first contribution is to provide insight on the (in)approximability of complex-valued knapsack problem considering active, reactive and apparent power capacity constraints. While many discrete optimization problems are computationally hard problems for computing the exact optimal solutions, approximate solutions that are close to the optimal solution may be efficiently computed by a polynomial-time algorithm. An effective class of approximation algorithms is constant approximation algorithms, which produce solutions that are within a constant factor from the optimal solution in polynomial time of the problem size. *Polynomial-time approximation scheme (PTAS)* can produce a solution that is within a factor $1 - \epsilon$ from the optimal solution for maximization problems (or $1 + \epsilon$ for minimization problems) for any $\epsilon > 0$ within time polynomial in the size of the input. The most effective class of approximation algorithms is the *fully polynomial-time approximation scheme (FPTAS)*, which additionally requires the running time to be polynomial in both the input size and $1/\epsilon$. *In this paper, we show the existence of a $(\frac{1}{2} - \epsilon)$ -approximation algorithm and the absence of FPTAS unless $P = NP$ for a class of complex-valued knapsack problems, considering non-negative active and reactive power demands.* We also apply our results to a setting of A/C power with moderate negative reactive power demands.

2. Truthfulness: Our second contribution is related to mechanism design. An algorithm is said to be *truthful* (or *incentive-compatible*) if none of the users has an incentive to misreport his true demand and utility. Namely, they will always truthfully reveal their private information. Truthful mechanisms are economically desirable in smart grid, because it simplifies the design of bidding strategies and facilitates users' adaptation. VCG based mechanisms [6], [7] can realize truthfulness, but require exact optimal solutions. However, many discrete optimization problems are computationally hard (e.g., knapsack problem). Therefore, we often need to adapt an

¹Purely resistive appliances have positive active power and zero reactive power. Appliances with capacitive or inductive components have non-zero reactive power, depending on the phase lag with the input power.

approximation algorithm to achieve truthfulness. Results in the literature [1], [5] give a sufficient condition for truthfulness in problem settings with single-minded users: the approximation algorithm satisfies a property known as *monotonicity*. But not all approximation algorithms satisfy monotonicity. *We show that our $(\frac{1}{2} - \epsilon)$ -approximation algorithm for basic complex-valued knapsack problem satisfies monotonicity, and hence leads to a truthful mechanism.*

II. PROBLEM FORMULATIONS AND DEFINITIONS

A. Real-valued One Dimensional Knapsack Problem

The traditional *knapsack problem* (1D-KS) is defined as:

$$(1D-KS) \quad \max_{x_k \in \{0,1\}} \sum_{k \in K} x_k u_k \quad (1)$$

$$\text{subject to} \quad \sum_{k \in K} x_k d_k \leq C \quad (2)$$

where

- 1) K is a set of users;
- 2) u_k is the utility of k -th user if its demand is satisfied;
- 3) d_k is the positive real-valued demand of k -th user;
- 4) C is a real-valued capacity of total satisfiable demand;
- 5) x_k is a decision of allocation determined by an algorithm ($x_k = 1$ means that k -th user's demand will be satisfied, otherwise $x_k = 0$).

This problem can model the discrete optimization of D/C electrical systems, where C is the capacity of D/C power. 1D-KS is NP-complete but has an FPTAS [2], [4].

B. Two Dimensional Knapsack Problem

In the setting of A/C electrical systems, a demand for power d can be expressed by a complex value $d = d^R + id^I$. One can impose two separate constraints on the capacity of total satisfiable demand of active power (i.e. the real part) and reactive power (i.e. the imaginary part). This problem can be formulated as a *two-dimensional knapsack problem* (2D-KS):

$$(2D-KS) \quad \max_{x_k \in \{0,1\}} \sum_{k \in K} x_k u_k \quad (3)$$

$$\text{subject to} \quad \sum_{k \in K} x_k d_k^R \leq C^R \text{ and } \sum_{k \in K} x_k d_k^I \leq C^I \quad (4)$$

2D-KS can be generalized to *m-dimensional knapsack problem* (m D-KS). m D-KS is NP-complete since 1D-KS is a special case. m D-KS only has a PTAS, and there is no FPTAS for m D-KS, when $m \geq 2$ (unless $P = NP$) [4].

C. Complex-valued Knapsack Problem

Our study concerns the capacity on apparent power (i.e. the magnitude of the total satisfiable demand). We formulate a complex-valued knapsack problem (C-KS) as follows:

$$(C-KS) \quad \max_{x_k \in \{0,1\}} \sum_{k \in K} x_k u_k \quad (5)$$

$$\text{subject to} \quad \left| \sum_{k \in K} x_k d_k \right| \leq C \quad (6)$$

where

- 1) $d_k = d_k^R + id_k^I$ is the *complex-valued* demand of power for k -th user;
- 2) C is a real-valued capacity of total satisfiable demand in apparent power.

Evidently, C-KS is also NP-complete, since 1D-KS is a special case when we set all $d_k^I = 0$.

We also define a general setting of knapsack problem for A/C electrical systems with a combination of capacity constraints on apparent power, active power and reactive power together. Namely, we formulate a general complex-valued knapsack problem (GC-KS) as follows:

$$(GC-KS) \quad \max_{x_k \in \{0,1\}} \sum_{k \in K} x_k u_k \quad (7)$$

subject to

$$\left| \sum_{k \in K} x_k d_k \right| \leq C \text{ and } \sum_{k \in K} x_k d_k^R \leq C^R \text{ and } \sum_{k \in K} x_k d_k^I \leq C^I \quad (8)$$

One of our contributions is to provide an approximation algorithm for C-KS and GC-KS.

D. Approximation Algorithm

Given an allocation $\{x_k\}$, we equivalently represent it by the satisfied subset of users $S \subseteq K$, where $S \triangleq \{k \in K : x_k = 1\}$. For a subset S , we denote $u(S) \triangleq \sum_{k \in S} u_k$.

We denote S^* to be an optimal solution of C-KS. An algorithm \mathcal{A} for C-KS is said to be a ρ -approximation for some constant $\rho(< 1)$, if S is the output of \mathcal{A} and $u(S) \geq \rho \cdot u(S^*)$ on each input.

For the convenience of analysis, in this paper we assume that each demand d_k lies in the first quadrant, such that

$$d_k^R \geq 0 \text{ and } d_k^I \geq 0 \quad (9)$$

We then provide a $(\frac{1}{2} - \epsilon)$ -approximation algorithm to solve C-KS under this assumption.

Note that our approximation algorithm is also applicable to the setting of negative reactive power, such that for all k

$$d_k^R \geq 0 \text{ and } d_k^I \geq |d_k^I| \quad (10)$$

The basic idea is to rotate the first quadrant by an angle of $\frac{\pi}{4}$ clockwise. Normally, positive active power demand implies a power consumer (who requires positive power from the power grid), while negative active power demand implies a power supplier. Hence, it is often that $d_k^R \geq 0$. For reactive power, inductors have positive reactive power, while capacitors have negative reactive power. In most electrical power systems regulations, it is often required that the absolute reactive power is not too large, as compared to active power [3]. Thus, the assumption in Eqn. (10) is justified in common settings.

E. Truthfulness

Suppose the set $\{(d_k, u_k)\}$ are users' private information. Without the precise knowledge of the private input, an algorithm cannot find a true utility-maximizing allocation. We seek to design a *truthful mechanism* in a way that incentivizes all users to report their true private information. A general approach is to enforce payment on each user that depends on reported inputs, such that revealing the true information maximizes his net utility² (utility minus the payment). In this way, no user has an incentive to misreport his private (d_k, u_k) .

²Here, to conform to the traditional usage of utility in knapsack problems, we change several important names in mechanism design: our net utility is the utility in mechanism design, and our utility is called valuation.

There is rich literature on designing truthful (or *incentive compatible*) mechanisms for problem settings well studied in traditional algorithmic design. The setting in C-KS is a special case of *single-minded* users. (See [1], [5])

For each user $k \in K$, we let $t_k = (d_k, u_k)$. Let $t = \{t_k\}$ and $t_{-k} = \{(d_j, u_j) : j \in K \setminus \{k\}\}$.

We define a *mechanism* as $\mathcal{M} = (\mathcal{A}, p)$, consisting of an algorithm $\mathcal{A}(t)$ that computes an allocation $\{x_k\}$ and payment $p_k(t)$ for each user k . We write $\mathcal{A}_k(t) = x_k$.

Since t_k is private information, user k may not truthfully reveal t_k . For any input of other users t_{-k} , the utility of user k depends on his report $t'_k = (d'_k, u'_k)$: if $\mathcal{A}_k(t) = 0$, he is not satisfied and utility equals 0; if $d'^R_k < d^R_k$ or $d'^I_k < d^I_k$, i.e., the power, if allocated to him, would not be enough for his demand, his utility is also 0; otherwise, he gets u_k .

Clearly, truth-telling results in net utility $U_k(t) = \mathcal{A}_k(t) \cdot u_k - p_k(t)$; if user k misreports t'_k , his net utility $U_k(t'_k, t_{-k})$ becomes the utility defined above minus $p_k(t'_k, t_{-k})$.

A *truthful* mechanism maximizes user k 's net utility when he reports true t_k for every t_{-k} , t_k and t'_k :

$$U_k(t_k, t_{-k}) \geq U_k(t'_k, t_{-k}) \quad (11)$$

An algorithm \mathcal{A} is said to be *truthful*, if there exists payments p such that mechanism $\mathcal{M} = (\mathcal{A}, p)$ is truthful.

A sufficient condition to ensure truthfulness for single-minded users is monotonicity [1], [5]. In our setting, monotonicity is specified as follows.

Definition 1: An algorithm \mathcal{A} is *monotone* if $\mathcal{A}_k(t) = 1$ implies $\mathcal{A}_k(t'_k, t_{-k}) = 1$ for any $u'_k \geq u_k$, $d'^R_k \leq d^R_k$ and $d'^I_k \leq d^I_k$.

Intuitively, in a monotone algorithm, if user k is satisfied with demand d_k and utility u_k , then he should be also satisfied when he has smaller demand and larger utility. The following theorem states the sufficiency of monotonicity [1], [5].

Theorem 1: [1], [5] Let \mathcal{A} be a monotone algorithm for single-minded users. Then there exists payment function p such that $\mathcal{M} = (\mathcal{A}, p)$ is truthful.

The basic idea is that, given a monotone algorithm \mathcal{A} , if we fix d_k and t_{-k} for user k , then \mathcal{A} defines a *critical value* $\theta_k(d_k, t_{-k})$, such that when u_k is above the critical value, k is satisfied; and when u_k is below the critical value, k is not satisfied. Then we can define a payment function $p(t)$, such that each satisfied user pays the critical value:

$$p_k(t) = \begin{cases} \theta_k(d_k, t_{-k}) & \text{if } \mathcal{A}_k(t) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

By Theorem 1, if we are able to design a monotone algorithm, we can transform it into a truthful mechanism. Moreover, the critical value for a given input can be computed in polynomial time by a binary search on interval $[0, u_k]$ for each user k during which we repeatedly test if k is satisfied by running algorithm \mathcal{A} . Therefore, a monotone polynomial time algorithm \mathcal{A} implies a polynomial time truthful mechanism.

While monotonicity usually holds for exact optimization, many well-known approximation algorithms violate monotonicity. One example is the FPTAS for 1D-KS [4], which is based on scaling utilities and applying the optimal pseudopolynomial dynamic programming algorithm, but is shown to be not monotone. Recently, [1] adapts it to a sophisticated scheme, which is a monotone FPTAS for 1D-KS.

III. MONOTONE $(\frac{1}{2} - \epsilon)$ -APPROXIMATION ALGORITHM

We present a polynomial-time monotone $\frac{1}{2}$ -approximation algorithm for C-KS, which relies on a polynomial-time monotone approximation algorithm for 1D-KS as a subroutine.

A. Basic Idea

We first present the intuition of our algorithm. Recall that we assume $d^R_k \geq 0$ and $d^I_k \geq 0$. Since we may preprocess the demands and eliminate every demand whose magnitude exceeds capacity C itself, without loss of generality, we assume all $|d_k| \leq C$.

Graphically, each demand is a vector $d = d^R + id^I$ in the first quadrant. A feasible solution of our problem is a subset of users whose sum of demands lies in region \mathcal{D} , the $1/4$ disk of radius C in the first quadrant. As shown in Fig. 1, \mathcal{D} is divided by chord PQ into triangle \mathcal{D}_1 and circular segment \mathcal{D}_2 . The $\frac{\pi}{4}$ line intersects chord PQ at point R .

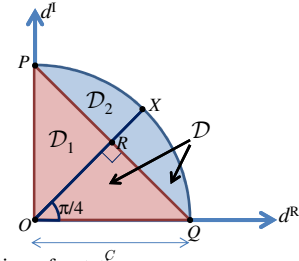


Fig. 1. An illustration of notations.

Our algorithm is based on the following two observations:

- 1) If we project all demands onto the $\frac{\pi}{4}$ line, i.e.,

$$\tilde{d}_k \triangleq (d^R_k + d^I_k)/\sqrt{2}, \quad (13)$$

we make all demands 1-dimensional. Now a subset of demands has sum $\sum \tilde{d}_k \leq C/\sqrt{2}$ (i.e., the sum vector does not go beyond point R on the $\frac{\pi}{4}$ line) if and only if its original sum vector $\sum d_k$ lies inside the triangle \mathcal{D}_1 . This is because that, the sum of projections, $\sum \tilde{d}_k$, is the projection of $\sum d_k$ on the $\frac{\pi}{4}$ line. Therefore, the subproblem on feasible region \mathcal{D}_1 can be solved by an algorithm for 1D-KS with demands changed to \tilde{d}_k and capacity to $C/\sqrt{2}$.

- 2) The subproblem on feasible region \mathcal{D}_1 is almost the whole story: First, an optimal solution in \mathcal{D} can contain at most one demand in \mathcal{D}_2 ; second, if an optimal solution consists of more than one demand, its sum can be broken into either two separate subsums lying in \mathcal{D}_1 , or, the sum of a vector in \mathcal{D}_2 and a subsum in \mathcal{D}_1 . Our algorithm essentially approximates the optimum between an optimal solution for the subproblem on feasible region \mathcal{D}_1 and an optimal solution on input demands lying in \mathcal{D}_2 . This gives an approximation ratio of $1/2 - \epsilon$.

B. Approximation Algorithm

We let $\text{Alg}^a[(d_k : k \in K), C]$ be our algorithm for C-KS, where $(d_k : k \in K)$ is the input of the demands and C is the capacity. Also, we let $\text{Alg}^{1d}[(d_k : k \in K), C]$ be a polynomial-time approximation algorithm for 1D-KS (for example, from [1]). We describe our algorithm as follows:

Algorithm 1 $\text{Alg}^a[(d_k : k \in K), C]$

```

1: for  $k \in K$  do
2:   Set  $\hat{d}_k = \min\{\frac{d_k^R + d_k^I}{\sqrt{2}}, \frac{C}{\sqrt{2}}\}$ 
3: end for
4: Set  $S = \text{Alg}^{1d}[(\hat{d}_k : k \in K), \frac{C}{\sqrt{2}}]$ 
5: Output  $S$ 

```

In Alg^a , we first project all demands onto the $\frac{\pi}{4}$ line. If the projection of a demand is larger than $C/\sqrt{2}$, we cut it off to $C/\sqrt{2}$. Then we use an approximation algorithm Alg^{1d} for 1D-KS to compute an allocation considering the projected demands and capacity $C/\sqrt{2}$.

Theorem 2: If Alg^{1d} is a ρ -approximation algorithm for 1D-KS, then Alg^a is a $\frac{\rho}{2}$ -approximation algorithm for C-KS.

Corollary 3: Since 1D-KS has an FPTAS [1], there is a $(\frac{1}{2} - \epsilon)$ -approximation algorithm for C-KS that runs in polynomial-time in the size of input and $1/\epsilon$, for any $\epsilon > 0$.

C. Proof of Theorem 2

Let S be the output of Alg^a and S^* be an optimal solution to C-KS, for which the feasible region is \mathcal{D} . We partition K into two disjoint sets K_1 and K_2 , such that $K_1 \triangleq \{k \in K : d_k \in \mathcal{D}_1\}$ and $K_2 \triangleq \{k \in K : d_k \in \mathcal{D}_2\}$. Note that the projection of any demand in K_1 onto the $\frac{\pi}{4}$ line is at most $C/\sqrt{2}$, whereas that in K_2 is larger than $C/\sqrt{2}$.

Let S_1 be the output of Alg^a , when the input is K_1 . Let S_2 be the output of Alg^a , when the input is K_2 . Let S_1^* and S_2^* be their corresponding optimal solutions. S_1^* is an optimal solution to the 1D-KS on projected demands within capacity $C/\sqrt{2}$, hence by our observations in Sec. III-A, is an optimal solution to the C-KS on feasible region \mathcal{D}_1 . On the other hand, since each demand in K_2 is changed to a vector exactly equivalent to the capacity limit of the 1D-KS, only one of them can be satisfied. Hence S_2^* chooses the one with maximum utility $S_2^* = \{\arg \max_{k \in K_2} u_k\}$.

Since any demand in K_2 will not combine with any in K_1 to form new feasible solutions to the 1D-KS, Alg^a outputs either a solution whose sum vector lies in \mathcal{D}_1 or a singleton set of a demand in K_2 , which is evidently a feasible solution to the C-KS.

Optimally the 1D-KS outputs $\arg \max\{u(S_1^*), u(S_2^*)\}$. Since Alg^{1d} is a ρ -approximation algorithm to the 1D-KS, we have $u(S) \geq \rho \cdot \max\{u(S_1^*), u(S_2^*)\}$.

Next, we analyze the approximation ratio of Alg^a in three cases. Here for a subset $S \subseteq K$, we denote

$$d(S) \triangleq \sum_{k \in S} d_k = \sum_{k \in S} d_k^R + \mathbf{i} \sum_{k \in S} d_k^I \quad (14)$$

Case (1): (ρ -approximation) We consider an optimal solution S^* , such that the sum of demands $d(S^*) \in \mathcal{D}_1$.

Proof: This is an easy case $u(S^*) = u(S_1^*)$. We have $u(S) \geq \rho \cdot \max\{u(S_1^*), u(S_2^*)\} \geq \rho \cdot u(S_1^*) = \rho \cdot u(S^*)$. ■

Case (2): ($\frac{\rho}{2}$ -approximation) We consider an optimal solution S^* , such that $d(S^*) \in \mathcal{D}_2$, and there exists a user $j \in S^*$ whose demand $d_j \in \mathcal{D}_2$.

Proof: Let $z \triangleq \sum_{k \in S^* \setminus \{j\}} d_k$. Thus, $d(S^*) = d_j + z$. That is, the sum of demands of S^* can be written as the sum of demand d_j and a demand subset sum z . (It is possible that S^* only consists of a single user j .) Note that $d_j \in \mathcal{D}_2$ and $z \in \mathcal{D}_1$. Otherwise, the projection of $d(S^*) = d_j + z$ on the $\frac{\pi}{4}$ line would exceed $2 \cdot C/\sqrt{2} > C$.

Moreover, we have $u(S^* \setminus \{j\}) \leq u(S_1^*)$, because S_1^* is an optimal solution for feasible region \mathcal{D}_1 . On the other hand, $u_j \leq u(S_2^*)$ by $S_2^* = \{\arg \max_{k \in K_2} u_k\}$. We obtain:

$$u(S^*) = u_j + u(S^* \setminus \{j\}) \leq u(S_2^*) + u(S_1^*) \quad (15)$$

Combining with $\max\{u(S_1^*), u(S_2^*)\} \geq \frac{1}{2}(u(S_1^*) + u(S_2^*))$ and $u(S) \geq \rho \cdot \max\{u(S_1^*), u(S_2^*)\}$, we get $u(S) \geq \frac{\rho}{2}u(S^*)$. ■

Case (3): ($\frac{\rho}{2}$ -approximation) We consider an optimal solution S^* , such that $d(S^*) \in \mathcal{D}_2$, and $d_k \in \mathcal{D}_1$ for every user $k \in S^*$.

Proof: First, we define $\tilde{d}(S) \triangleq \sum_{k \in S} \tilde{d}_k$. By projection on the $\frac{\pi}{4}$ line, it is equivalent to saying that $\tilde{d}(S^*) > C/\sqrt{2}$, and $\tilde{d}_k \leq C/\sqrt{2}$ for every user $k \in S^*$.

The proof of Case (3) is immediate from Lemma 1 as follows: By Lemma 1, we have $\tilde{d}(T) \leq C/\sqrt{2}$ and $\tilde{d}(S^*/T) \leq C/\sqrt{2}$ for some subset $T \subseteq S^*$. That is, $d(T) \in \mathcal{D}_1$ and $d(S^*/T) \in \mathcal{D}_1$.

Thus, $u(T) \leq u(S_1^*)$ and $u(S^*/T) \leq u(S_1^*)$. Moreover, $u(S^*) = u(T) + u(S^*/T)$. Hence, $u(S^*) \leq 2u(S_1^*)$.

Finally, the total utility of the output solution of Alg^a ,

$$u(S) \geq \rho \cdot \max\{u(S_1^*), u(S_2^*)\} \geq \rho \cdot u(S_1^*) \geq \frac{\rho}{2}u(S^*) \quad (16)$$

Combining Cases (1)-(3): $\min\{\rho, \rho/(1+\rho), \rho/2\} = \rho/2$, we complete the proof of the approximation ratio of Alg^a as $\rho/2$.

Lemma 1: For a set of n positive real-valued numbers a_1, \dots, a_n satisfying $\sum_{i=1}^n a_i \leq C$, $a_i \leq C'$ for all i and $C' \geq C/\sqrt{2}$, there exists a subset $T \subseteq \{1, \dots, n\}$ such that

$$\sum_{i \in T} a_i \leq C' \quad \text{and} \quad \sum_{i \in \{1, \dots, n\} \setminus T} a_i \leq C' \quad (17)$$

Proof: Let j be the smallest index such that the partial sum exceeds C' , i.e., $\sum_{i=1}^{j-1} a_i \leq C'$ and $\sum_{i=1}^j a_i > C'$. Clearly $j \geq 2$ since all $a_i \leq C'$.

Let $x = \sum_{i=1}^{j-1} a_i$, $z = a_j$ and $y = \sum_{i=j+1}^n a_i$.

Note that $\sum_{i=1}^n a_i = x + y + z$. We already have

$$x \leq C', \quad z \leq C', \quad x + y + z > C' \quad \text{and} \quad x + z > C' \quad (18)$$

The lemma holds if $y + z \leq C'$, because we can set $T = \{1, \dots, j-1\}$.

If $y + z > C'$, then we obtain:

$$x + y = 2(x + y + z) - (x + z) - (y + z) \quad (19)$$

$$< 2C - 2C' \leq (2 - \sqrt{2})C < \frac{C}{\sqrt{2}} \leq C' \quad (20)$$

because $x + y + z \leq C$. Hence, we can set $T = \{1, \dots, j - 1, j + 1, \dots, n\}$. ■

D. Monotonicity

We also show that our algorithm Alg^a satisfies monotonicity.

Theorem 4: There is a truthful $(1/2 - \epsilon)$ -approximation algorithm for C-KS that runs in polynomial time in the size of the input and $1/\epsilon$, for any $\epsilon > 0$.

Proof: Note that 1D-KS has a monotone FPTAS from [1]. We use it as Alg^{1d} . Theorem 4 follows immediately from Theorem 1 and Corollary 3, if we prove that Alg^a is monotone.

We then need to show that, if user k is satisfied by Alg^a with demand d_k and utility u_k , k is also satisfied with demand d'_k and utility u'_k , where $u'_k \geq u_k$ and $d_k^R \leq d'_k^R$ and $d_k^I \leq d'_k^I$, while all inputs of other users do not change.

User k is satisfied by Alg^a on d'_k and u'_k if and only if it is satisfied by Alg^{1d} on \hat{d}'_k and u'_k . Since $\hat{d}_k = \min\{\frac{d_k^R + d_k^I}{\sqrt{2}}, \frac{C}{\sqrt{2}}\}$, $d_k^R \leq d'_k^R$ and $d_k^I \leq d'_k^I$ implies that $\hat{d}'_k \leq \hat{d}_k$. Then from the monotonicity of Alg^{1d} , k is satisfied by Alg^{1d} , and hence by Alg^a . ■

IV. INAPPROXIMABILITY

In this section, we complete the study of C-KS by providing an inapproximability result. We show that C-KS can not have FPTAS, unless $P = NP$.

We remark that it is known there is no FPTAS for 2D-KS (see [4]), which does not have direct implications for C-KS. Neither can the reduction in its proof be directly adapted. Our proof is a clever modification of the reduction for 2D-KS.

We reduce the EQUIPARTITION problem to C-KS:

Definition 2: (EQUIPARTITION Problem): Given a set of positive integers $\{w_k : k \in K\}$, with $|K| = n$ even, we determine if there is a subset of items $S \subseteq K$ such that

$$|S| = \frac{n}{2} \text{ and } \sum_{k \in S} w_k = \sum_{k \notin S} w_k$$

It is well-known that EQUIPARTITION is NP-complete.

Theorem 5: There is no FPTAS for C-KS, unless $P = NP$.

Proof: We define a decision version of C-KS with a cardinality objective: given $\{w_k : k \in K\}$, a capacity bound C and a cardinality bound M , we determine if there is a subset of items S such that

$$|S| \geq M, \text{ and } \left| \sum_{k \in S} d_k \right| \leq C$$

Now we map every instance of EQUIPARTITION to an instance of the C-KS decision problem that always yields the same answer.

Given $\{w_k : k \in K\}$ from EQUIPARTITION, define

$$M = n/2, \quad d_k^R = w_k, \quad d_k^I = \beta(w_{\max} - w_k),$$

$$C = \sqrt{\left(\frac{W}{2}\right)^2 + \beta^2 \left(\frac{nw_{\max}}{2} - \frac{W}{2}\right)^2}$$

where $W \triangleq \sum_{k=1}^n w_k$, $w_{\max} \triangleq \max\{w_k : k \in K\}$. Note that in our reduction, $d_k^I \geq 0$.

As shown in Fig. 2, the feasible region \mathcal{D} for C-KS is the $\frac{1}{4}$ disk of radius C in the first quadrant. Since for any subset $S \subseteq K$, $\sum_{k \in S} d_k^I = \beta(|S| \cdot w_{\max} - \sum_{k \in S} d_k^R)$, the cardinality constraint $|S| \geq \frac{n}{2}$ imposes all solutions to have its sum vector in the halfplane $H : d^I \geq \beta(\frac{nw_{\max}}{2} - d^R)$. The dividing line of H goes through point $P : (\frac{W}{2}, \beta(\frac{nw_{\max}}{2} - \frac{W}{2}))$. Our main idea is to set $\beta > 0$ such that the dividing line of H coincides with the tangent line at P . Thus we make the intersection of H and \mathcal{D} exactly P , which implies $|S| = \frac{n}{2}$ and $\sum_{k \in S} w_k = \frac{W}{2}$ for any solution S to our reduced C-KS decision problem instance.

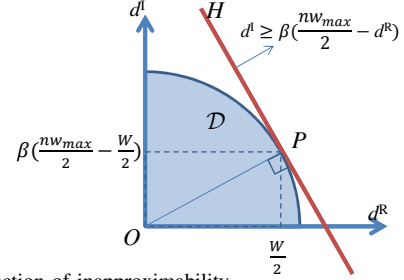


Fig. 2. Reduction of inapproximability.

On the other hand, it is clear that each subset S satisfying conditions of EQUIPARTITION also satisfies conditions of the reduced C-KS decision problem. Therefore, the solution of the reduced C-KS decision problem is equivalent to the solution of EQUIPARTITION.

To determine a proper β , since the dividing line of halfplane H goes through P , it coincides with the tangent line at P if and only if they have the same slope, i.e.,

$$-\frac{\frac{W}{2}}{\beta(\frac{nw_{\max}}{2} - \frac{W}{2})} = -\beta.$$

Solving the above equation, we obtain

$$\beta = \sqrt{\frac{W}{nw_{\max} - W}},$$

which is > 0 unless all weights are equal. In this case, we set $\beta = 0$. It is trivial to check that both instances have answer "Yes".

So far we have shown the NP-completeness of the complex valued knapsack decision problem. So its maximization version, where $|S| \geq M$ is replaced by $\max |S|$, is NP-hard. This is equivalent to the original complex-valued problem with all $u_k = 1$.

Finally, we use the standard technique to prove the inapproximability by FPTAS (for the above maximization version). Suppose that there exists an FPTAS for any $\epsilon > 0$ in time polynomial in n and $1/\epsilon$. Then we choose $\epsilon = \frac{1}{n+1}$. Let the sum of values of the optimal solution be $z^* > 0$ and that of the approximation solution produced by FPTAS be z^A . We obtain:

$$z^A \geq (1 - \epsilon)z^* > z^* - z^*/n \geq z^* - 1$$

because $z^* \leq n$. Moreover, since z^* is an integer, this implies that the FPTAS can solve the problem exactly in polynomial time, contradicting the NP-hardness of the problem. Therefore, there is no FPTAS to C-KS. ■

V. GENERALIZED COMPLEX-VALUED KNAPSACK

We are also able to solve the generalized problem GC-KS, by changing our approximation algorithm Alg^a in Section III. Now, instead of an approximation algorithm Alg^{1d} for 1D-KS as a subroutine, we rely on an approximation algorithm for 3D-KS (three dimensional knapsack problem) as a subroutine. 3D-KS is defined as:

$$(3D\text{-KS}) \quad \max_{x_k \in \{0,1\}} \sum_{k \in K} x_k u_k \quad (21)$$

subject to

$$\sum_{k \in K} x_k d_k^1 \leq C^1 \text{ and } \sum_{k \in K} x_k d_k^2 \leq C^2 \text{ and } \sum_{k \in K} x_k d_k^3 \leq C^3 \quad (22)$$

where $d_k^j \geq 0$ for $j = 1, 2, 3$.

A. Basic Idea

Now a feasible solution of our problem is a subset of users whose sum of demands lies in the intersection of halfplanes $d^R \leq C^R$, $d^I \leq C^I$, and the $1/4$ disk of radius C in the first quadrant. In the most general case ($C^R, C^I < C$), both halfplanes cut the circle, which also cut the original regions \mathcal{D}_1 and \mathcal{D}_2 in Section III. Fig. 3 shows the new \mathcal{D}_1 (polygon $PSTQO$) and \mathcal{D}_2 . Clearly, the feasible region \mathcal{D} is the union of \mathcal{D}_1 and \mathcal{D}_2 .

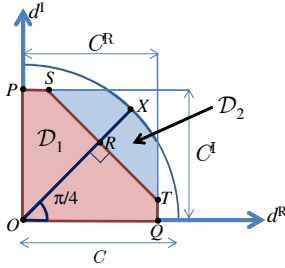


Fig. 3. An illustration of notations.

Recall that OR is perpendicular to ST and the length of OR is $C/\sqrt{2}$. If we denote the projection of a demand d_k onto line OR by \tilde{d}_k , the region \mathcal{D}_1 corresponds to 3-dimensional linear constraint $\sum_{k \in K} x_k \tilde{d}_k \leq C/\sqrt{2}$, $\sum_{k \in K} x_k d_k^R \leq C^R$ and $\sum_{k \in K} x_k d_k^I \leq C^I$. Thus the subproblem on feasible region \mathcal{D}_1 can be solved by a 3-dimensional knapsack algorithm.

On the other hand, the solutions in polygon \mathcal{D}_1 is almost the whole story by the same reason as in Section III. Here our algorithm directly takes the maximum between an optimal solution for the subproblem on feasible region \mathcal{D}_1 and an optimal solution on input demands lying in \mathcal{D}_2^3 . This gives an approximation ratio of $1/2 - \epsilon$.

The degenerate cases ($C^R \geq C$ or $C^I \geq C$ or both) can be treated easily by setting T, Q to be the intersection point of the circle and the d^R -axis, or setting P, S to be the intersection point of the circle and the d^I -axis, or both.

B. Approximation Algorithm

Let $\text{Alg}^c[(d_k : k \in K), C, C^R, C^I]$ be our approximation algorithm for GC-KS. Let $\text{Alg}^{3d}[(d_k^1, d_k^2, d_k^3) : k \in K], C^1, C^2, C^3]$ be an approximation algorithm for 3D-KS (e.g., from [2] or [4]). We describe our approximation algorithm to GC-KS by Alg^c .

³We can also modify Algorithm 1 of Section III in this way, which is more intuitive and improves the ratio of case (2) to $\rho/(1 + \rho)$, but has no monotonicity guarantee. Since we are not aware of any monotone PTAS for 3D-KS, we take this alternative here.

Algorithm 2 $\text{Alg}^c[(d_k : k \in K), C, C^R, C^I]$

```

1: for  $k \in K$  do
2:   Set  $\tilde{d}_k = \frac{d_k^R + d_k^I}{\sqrt{2}}$ 
3: end for
4: Set  $S_1 = \text{Alg}^{3d}[(\tilde{d}_k, d_k^R, d_k^I) : k \in K], \frac{C}{\sqrt{2}}, C^R, C^I]$ 
5: Set  $S_2 = \{\arg \max_{k \in K: d_k \in \mathcal{D}_2} \{u_k\}\}$ 
6: Set  $S = \arg \max_{S_1, S_2} \{u(S_1), u(S_2)\}$ 
7: Output  $S$ 

```

Theorem 6: If Alg^{3d} is a ρ -approximation algorithm for 3D-KS, Alg^c is a $\frac{\rho}{2}$ -approximation algorithm for GC-KS.

Corollary 7: Since 3D-KS has a PTAS [4], there is a $(\frac{1}{2} - \epsilon)$ -approximation algorithm for GC-KS that runs in polynomial-time in the size of input, for any $\epsilon > 0$.

The proof of Theorem 6 is similar to that of C-KS. We omit the proof here because of the space limit. Note that here $u(S_2) = u(S_2^*)$ since S_2 only consists of a single demand in \mathcal{D}_2 that has maximum utility. And $u(S) = \max\{u(S_1), u(S_2)\} \geq \max\{\rho \cdot u(S_1^*), u(S_2^*)\}$.

VI. CONCLUSION

Balancing the supplies and demands in smart grid becomes increasingly challenging, which highlights the importance of our fundamental study of discrete resource allocation in A/C electrical systems. We investigate a basic problem underpinning smart grid operations, *complex-valued knapsack problem*, a departure from the traditional real-valued resource allocation settings. Our results of (in)approximability and truthfulness provide the basic insights on efficient and economically viable mechanisms for discrete optimization in A/C electrical systems. We envision a wide range of applications that build on our results, such as (1) scheduling of EV charging with inelastic commuting distance constraints, (2) scheduling of heavy industry productions that require large minimum power supplies. In the future, we will work on the extension of our results to a networked setting, which is a generalized integer multi-commodity problem with complex-valued supplies and demands. We also seek to find a PTAS for our problem, closing the gap between constant approximation and FPTAS.

REFERENCES

- [1] P. Briest, P. Krysta, and B. Vocking. Approximation techniques for utilitarian mechanism design. In *STOC'05*, 2005.
- [2] A. Frieze and M. Clarke. Approximation algorithm for the m-dimensional 0-1 knapsack problem. *European Journal of Operational Research*, 15:100–109, 1984.
- [3] J. Grainger and W. Stevenson. *Power System Analysis*. McGraw-Hill, 1994.
- [4] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2010.
- [5] D. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. In *EC'99*, 1999.
- [6] N. Nisan. Introduction to mechanism design. In *Algorithmic Game Theory*, chapter 9, pages 209–242. 2007.
- [7] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.